Ain Shams University

Faculty of Computer and Information Science

Income Prediction

Habiba Alaa Eldin Adel Elsanadidy	2021170161
Rana Wahid Tammam Othman	2021170195
Fatma Gamal Saleh Mohamed Gad	2021170386
Habiba Bakr Ahmed Tammam	2021170159
Amna Ahmed Mirghani	2021170087
Wessal Salah	2021170614
Khira Ebe	2021170178

Income Prediction

Introduction

This project aims to predict individuals' income levels based on various demographic and employment-related features using machine learning algorithms. Different models and different approaches have been explored in order to achieve the best performance, and to provide insight into the factors that contribute to one's income level.
A possible application of this model would be to help non-profit organizations evaluate their much-needed donation requests from different individuals.

Dataset

The train dataset contains 32560 rows, and 14 different independent features. The goal is to predict whether a person, based on their features, earns more than 50K$ per year or not. Since the prediction can be one of two values (>50K or <=50K), this is a binary classification problem.

Exploratory data analysis

1. **Overview of the train data, its dimensions and its data types**
2. **Data cleaning**
    - No explicit null values are present in the data.
    - There are outliers. However, attempts at handling them resulted in a worse performing model, so they were left unaltered.
    - "Education" column is dropped because it gives information already given by "education_num".

- “Fnlwgt” is also dropped to avoid confusing the model, as it is a somewhat controversial feature of the dataset. Duplicates are found and removed.
- Wrong data in the form of ‘?’ is present in several columns, making up a small percentage. They are replaced by the mode of their column.

3. **Data visualization**
    - Pair plots to get insight into the columns.
    - Graphs for each column showing the count of each value and how it affects the target column.
    - Box plots to show outliers and how they could be handled.
4. **Feature selection**
    - The values in “marital-status” are reduced to make things easier for the model.
    - Chi-square test is used to see the correlation between columns, and what are the features affecting the target.
5. **Data preprocessing**

- Categorical columns are encoded using one-hot encoding and label encoding.
- Standardization was attempted, but it decreased the accuracy, so it was skipped.

6. **Test data cleaning and preparation**

Modeling

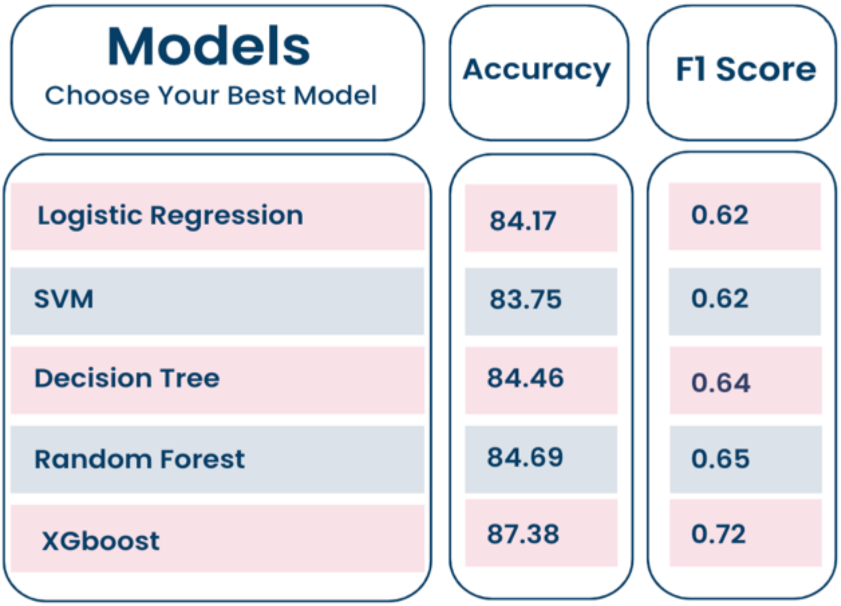
- **Logistic Regression**
  A statistical model used for binary classification problems, where the goal is to predict the probability of an event or outcome belonging to one of two classes. It is a popular and widely used algorithm in machine learning and statistics. The logistic regression model uses the logistic function (also known as the sigmoid function) to model the relationship between the features (input variables) and the probability of

the outcome. The logistic function maps any real-valued number to a value between 0 and 1, representing a probability.

- **SVM**
  The SVC (Support Vector Classifier) is a popular algorithm used for classification tasks in machine learning. It belongs to the family of supervised learning algorithms and is based on the concept of support vector machines (SVMs). The SVC algorithm works by finding an optimal hyperplane in a high-dimensional feature space that separates the data points of different classes. The goal is to maximize the margin between the classes, which is the distance between the hyperplane and the closest data points of each class.
- **Decision Tree**
  It is a non-parametric supervised learning method that builds a flowchart-like structure, where each internal node represents a feature or attribute, each branch represents a decision rule, and each leaf node represents the outcome or prediction. The decision tree algorithm recursively partitions the training data based on the values of different features, aiming to create homogeneous subsets at each level that are as pure as possible in terms of the target variable (for classification tasks). This partitioning process continues until a stopping criterion is met, such as reaching a maximum depth or a minimum number of samples in a leaf node.
- **Random Forest**
  Random Forest is a popular ensemble learning algorithm that combines multiple decision trees to improve prediction accuracy and reduce overfitting. It is widely used for both classification and regression tasks in machine learning. The Random Forest algorithm works by creating a collection of decision trees, where each tree is trained on a different subset of the training data, randomly selected with

replacement. Additionally, at each split in the tree, only a random subset of features is considered for determining the best split.

- **XGBoost**
  It is designed to handle both classification and regression tasks and can work with a variety of data types, including numerical and categorical features. It builds an ensemble of weak prediction models, typically decision trees, in a sequential manner, where each new model corrects the mistakes made by the previous models.



Models Choose Your Best Model	Accuracy	F1 Score
Logistic Regression	84.17	0.62
SVM	83.75	0.62
Decision Tree	84.46	0.64
Random Forest	84.69	0.65
XGboost	87.38	0.72

As shown above, XGBoost yielded the best accuracy and F1 value. The following **hyperparameters** were tweaked to get a higher accuracy:

- *n_estimators:* It refers to the number of boosting rounds or decision trees to be built. Increasing the number of estimators may improve the model's performance, but it can also lead to overfitting if set too high. It's a crucial parameter to tune as it balances model complexity and computation time.

- *learning_rate:* It determines the step size at each boosting iteration. A smaller learning rate makes the model converge slowly but can potentially yield better results by allowing more fine-grained adjustments. On the other hand, a larger learning rate makes the model converge faster but may result in overshooting the optimal solution.
- *max_depth:* It sets the maximum depth of each decision tree. A higher value allows the tree to capture more complex interactions in the data, but it can also lead to overfitting. It's important to find an optimal value that balances the model's capacity to learn complex patterns without compromising generalization.
- *gamma:* It specifies the minimum loss reduction required to split a node further. A higher gamma value makes the algorithm more conservative, as it requires a higher loss reduction to make additional splits. This parameter can help control the complexity of the tree and prevent overfitting.

Conclusion

By setting the following values for the hyperparameters: *(n_estimators=700, learning_rate=0.03, max_depth=5, gamma=0.01),* the highest accuracy <u>87.38 %</u> was achieved.